MODELLING AND NONLINEAR CONTROL OF A QUADCOPTER FOR STABILIZATION AND TRAJECTORY TRACKING

**Abstract -** A Quadcopter is an Unmanned Aerial Vehicle (UAV) that is capable of vertical take-off and landing. This thesis presents the mathematical modeling and control of a nonlinear quadcopter system for stabilization and trajectory tracking. The mathematical model of the system dynamics of the quadcopter is derived using Newton and Euler equations with proper references to the appropriate frame or coordinate system. A PID control algorithm is developed for the nonlinear system for stabilization. Another nonlinear control technique called Feedback Linearization (FBL) using Nonlinear Dynamic Inversion (NDI) is discussed and implemented on the quadcopter system. The complete derivation of the full state FBL system using NDI is also shown in this paper. The proposed control algorithms are implemented on the quadcopter model using MATLAB and analyzed in terms of system stabilization and trajectory tracking. The PID controller produces satisfactory results for system stabilization but the FBL system performs better for trajectory tracking of the quadcopter system.

**Keywords** - Quadcopter, Control, Nonlinear control, PID Controller, Nonlinear Dynamic Inversion, Feedback Linearization.

**I Introduction**

A quadcopter, also known as a quadrotor, is a rotor-based aerial vehicle. It's a multi-rotor aircraft propelled by four rotors. To balance the torque, these rotors are built with two pairs of opposite rotors revolving clockwise and the other rotor pair moving anti-clockwise. A quadcopter's dynamics are extremely nonlinear, it is an underactuated system with six degrees of freedom and four control inputs which are the rotor velocities. The quadcopter is controlled by adjusting the rotors' angular speeds, which alters the quadcopter's torque and thrust characteristics. The quadcopter is designed with four rotors in cross-configuration as seen in Figure 1. Two opposite rotors rotate in the same direction, the altitude and position of the quadcopter can be controlled by varying the rotor’s angular speed. The quadcopter can be kept in a balanced position without spinning if the generated torque of the motors T1, T2, T3, and T4 are the same.

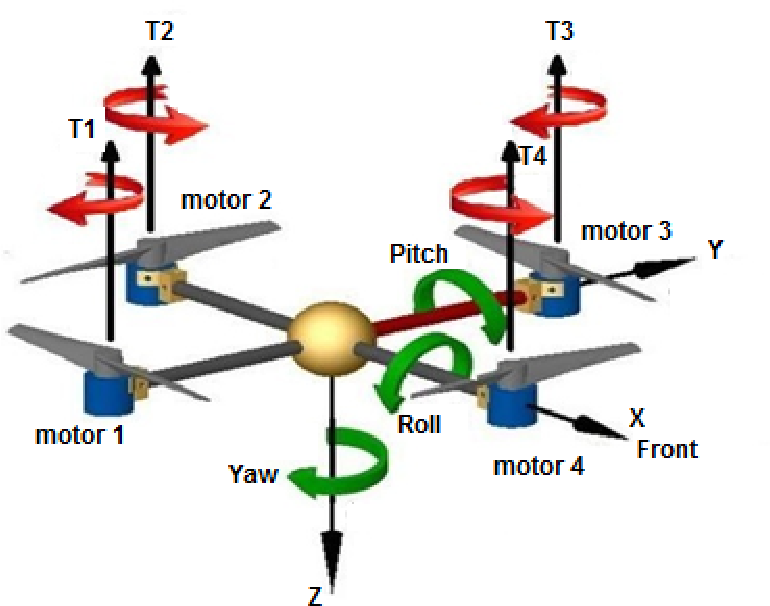


Figure 1: Quadcopter Dynamics [1]

Thrust controls the altitude of the quadcopter for ascending and descending and is achieved by increasing or reducing the rotational speed of motors 1, 2, 3 and 4 simultaneously. Roll is movement of the quadcopter by tilting either left or right to allow side movements. Pitch is movement of the quadcopter by tilting either front or back to allow forward or backward movement. Yaw is movement in a clockwise or anticlockwise manner while staying level to the ground to change the direction of the quadcopter. These flight motions can be achieved by controlling the rotational speeds of the four motors

Quadcopter designs have recently been popular in unmanned aerial vehicle research (UAVs). To control and stabilize the aircraft, these UAVs use an electrical control system and sensors [2]. Small Unmanned Aerial Vehicles (sUAV) have become a reality thanks to recent advancements in microcomputer technology, sensor technology, control systems, and dynamics theory. Due to their compact size, low cost, and agility, these systems are being used in a range of applications. As the range and complexity of applications for quadcopters expands on a daily basis, the control techniques utilized on the system must also improve. This paper aims to develop a mathematical model of a quadcopter system and implement nonlinear control techniques on the derived model for stabilization and trajectory tracking of a quadcopter.

**II Literature Survey**

In the 1920s and 1930s, a few manned designs existed. These vehicles were among the first heavier-than-air vertical take-off and landing (VTOL) vehicles to be successfully tested [2]. However, due to weak stability augmentation and restricted control authority, early prototypes performed poorly, while later versions required too much pilot effort. Historically, basic linear control techniques were used to ensure easy computation and flight stability but due to improved modelling techniques and faster computational capabilities, it is now possible to run comprehensive nonlinear techniques on real-time [3]. Linear control techniques can be implemented on quadcopter systems by linearizing the system about an equilibrium point but such approximation may not preserve the dynamics of the system at every point and are not usually very effective in practical scenarios.

A comprehensive mathematical modeling of the kinematics and dynamics of a quadcopter system is described in [1]. The kinematic aspect of the modeling depicts the quadcopter motion without taking into account the forces acting on it whereas the dynamics explains the forces causing the motion. This gives an overview of the response of quadcopters when under the influence of extraneous forces. Nonlinear state equations for the system were derived and the manual method to obtain minimum steady state error for PID tuning is presented. This research resulted in a quadcopter with a composite frame, assembled and implemented in real time for pitch, roll and stabilization with PID control. The results yield adequate performance when the hardware configuration is at the hover point.

The mathematical modelling and control of a quadcopter is presented in [4]. This paper shows the basics of quadcopter modelling and control to serve as a stepping stone for future research advancement. A detailed study of the mathematical model of the quadcopter dynamics is given and the differential state equations of the system dynamics is derived from both the Newton-Euler and Euler-Lagrange equations. A simple mathematical model is presented in this paper neglecting several aerodynamic effects and the modelling of the electric motor spinning the rotors. The developed model is simulated to analyze the behavior of the system and control algorithms are implemented for stability and trajectory control of the system. The PD controller is used to achieve this and a heuristic method is developed for trajectory control of the quadcopter flight. An integration of the PD controller and the heuristic method is used to minimize the disturbance on the quadcopter system caused by external forces. The PD control algorithm was integrated into the heuristic method because the heuristic method developed did not account for unmodelled disturbances (wind). A decline in performance of the PD controller was observed if the parameter values are extremely small or high. The necessity for an actual experimental prototype was identified, so that realistic and accurate findings could be acquired and comparison between the simulated results and real-life measurements would be possible.

A review of control algorithms for autonomous quadrotors is presented in [5]. Several control algorithms are analyzed on the system to propose hybrid systems which would combine the advantages from more than one control algorithm. The focus of this paper was to implement the optimal control algorithm of the quadrotor manned aerial vehicle for game counting in the protected game reserves in Africa. The control algorithms analyzed in this paper are: PID, Linear Quadratic Regulator (LQR), Sliding mode control, Backstepping control, Adaptive control and Artificial neural networks. From the review, no particular control algorithm gives the best performance in all required features in terms of fast response, disturbance rejection, stability, robustness, adaptability, optimality and tracking ability. A hybrid control system could have the best combination of some of these features but doesn’t guarantee an overall optimum performance. Certain features are more important depending on the application of the quadrotors hence compromises would have to be made to implement control algorithms that excel in such areas of application. A quadrotor to be used for game counting is designed prioritizing certain characteristics such as high endurance, high agility, high cruising, low noise and vertical take-off and landing ability. This paper concluded that the designed quadrotor would be suitable for assisting nature conservationist in game counting and obtaining accurate animal statistics.

A thesis on quadcopter flight mechanics model and control algorithms is shown in [6]. The focus of this thesis was to develop a nonlinear model of a quadcopter flight mechanics with suitable control algorithms for stabilization and implement it in MALTLAB/Simulink. The dynamics of the quadcopter were derived from the equation of motion and forces and a nonlinear model was obtained based on these dynamics. Aerodynamic effects such as non-zero free steam and blade-flapping are ignored in the derivation of the system equations but other forces such as air friction and drag forces are considered. For analysis of the system, the model is linearized at a stable hover point and both the linear and nonlinear models are analyzed. The Gradient Descent Method is used for controller tuning to obtain optimum control parameters and the system is simulated for a given period of time. A better performance was observed from the automatically tuned PID controller (using the Gradient Descent method) than the manually tuned PID controller.

The design and control of quadrotors in relation to autonomous flying [2] is a thesis that entails the modelling, design and control of small flying drones with a focus on Vertical Take-Off and Landing (VTOL) systems. In this thesis, an autonomous quadrotor called OS4 is modelled and simulated with various controllers. A more complex mathematical model is presented here as the system dynamics accounts for more realistic aerodynamic coefficients along with sensor and actuator models. The OS4 quadcopter model is simulated with five different control algorithms. The first one was based on Lyapunov theory and was implemented for attitude stabilization but after simulation on OS4, it was observed that it was not stable enough to allow hover flight. The second control algorithm implemented was the PID controller and it was suitable for attitude stabilization of the quadcopter system when flying near hover but this was only possible in the absence of huge disturbances. The third control algorithm was an LQR controller which presented adequate stabilization but this control algorithm was observed to be less adaptive than the PID. The fourth control algorithm was the Backstepping controller which achieved excellent stabilization and control in the presence of relatively large disturbances. The fifth control algorithm implemented was the Sliding-mode technique but this control algorithm did not give desirable results because the switching characteristics of the controller was not compatible with the dynamics of the quadcopter system. After the implementation and analysis of these five control algorithms, a suitable control algorithm was developed by the integration of the PID and the Backstepping algorithm called Integral Backstepping. This was used for altitude, attitude, position and trajectory control and the implementation of this control algorithm on the OS4 enabled the system to take-off, hover, land and avoid collisions. The thesis noted that the OS4 was the first collision avoidance quadcopter system.

**III Methodology**

**A Mathematical Modelling**

A quadcopter is an aerial vehicle with four rotors that enable motion in different directions. Proper explanation of the system dynamics would require comprehension of the 6-degrees of freedom concept which presents the position and orientation of the quadcopter in three dimensions (3-D).

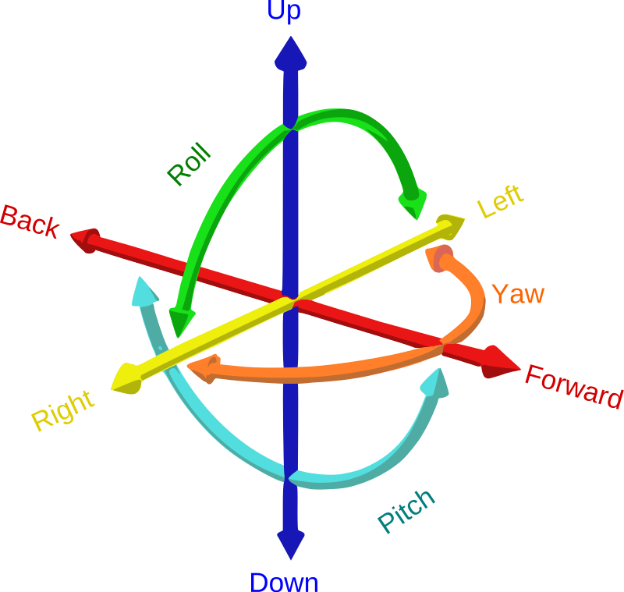


Figure 2: Six Degrees of Freedom [taken from Wikipedia]

The 6-degrees of freedom describes the position of a body with six coordinates categorized in two reference frames. The first reference frame is the fixed coordinate system known as the inertial or earth frame which is depicted by the x, y, and z coordinates in the cardinal points of North, East and Down. The second reference frame is a mobile coordinate system known as the body frame which is depicted by the angles with respect to the body center of gravity. The quadcopter is an underactuated nonlinear system because it has four inputs and six outputs. The system is complex and in order to control it, the quadcopter is modelled on the following assumptions[2]:

1. The structure is rigid
2. The structure is axis symmetrical
3. The Centre of Gravity and the body fixed frame origin coincide
4. The propellers are rigid
5. Thrust and drag are proportional to the square of the propeller’s speed

**Euler Angles**

The Euler angles are three angles introduced by Leonhard Euler to describe the orientation of a rigid body in a coordinate system. They are also used to describe the relationship between two different reference frames and to convert the coordinates of a point in one reference frame to coordinates of the same point in another reference frame. Euler angles are denoted as for the roll, pitch and yaw angles respectively and represent the rotations of a body about the axes of a coordinate system.

Any orientation of a rigid body can be achieved by combination of the three basic Euler angles. The rotation matrices are given by [3]:

where . The rotation matrix depicting the relationship between the inertial frame and the body frame is given as:

R is a rotational matrix and is orthogonal such that .

**Reference Frame Transformation**

Assume to be a vector of linear and angular positions in the inertial frame and let be a vector of linear and angular velocities in the body frame.

Typically, the derivative of angular positions should give angular velocities but the angular positions and velocities above are in a different frame so we require some transformation matrix to convert from one reference frame to another.

From this statement, , instead:

Conversely,

**Rotational Motion**

Assuming the quadcopter is a rigid body and using Euler’s equations for rigid bodies, the dynamics equation in the body frame is given as:

We also assume that the quadcopter has a symmetric structure with the four arms aligned with the body x and y-axes. Therefore, the inertia matrix is a diagonal matrix in which .

The gyroscopic forces are caused by the combined rotation of the four rotors and the quadcopter body such that:

In matrix form,

The external torque,

The roll torque component , and the pitch torque component , are obtained from standard mechanics where and motors are arbitrarily chosen to be on the roll-axis while and are arbitrarily chosen to be on the pitch-axis.

For the yaw-axis, the rotor axis is pointing in the z-direction in the body frame, the torque created around the rotor axis is given by:

where is positive for the propeller if the propeller is spinning clockwise and negative if it is spinning counter-clockwise. The term can be ignored because in steady state,

Therefore, the total torque about the z-axis is given by the sum of all the torques from each propeller:

Therefore, the torque matrix can be written as:

where is the thrust coefficient, is the drag coefficient and is the distance between rotor and the center of mass of the quadcopter.

**Translational Motion**

Using Newtonian equation to model the linear dynamics, the extraneous forces acting on the quadcopter is given as:

Drag force , is the force acting on the system as a result of fluid friction (air resistance). A simplified equation form is used where friction is modelled as being proportional to the linear velocity in all directions.

The angular velocity of the rotor creates a force in the direction of the rotor axis (z-direction). The combined forces create thrust in the direction of the body z-axis such that:

Since it acts in the z-axis:

**State Space Model**

From the rotational dynamics discussed, equation 11 can be rewritten as:

From the linear dynamics discussed, equation 21 can be written out as:

**B Quadcopter System Stabilization**

A quadcopter has only four control inputs and primarily six outputs of interest which makes it an underactuated system. This is resolved by separating it into two different control loops, with one working with attitude states and the other with position states. The quadcopter's angular motion is independent of the linear components, but the linear motion is determined by the angle variables. As a result, the goal is to control the attitude variable, which is independent of the linear motion, and then control the linear motion. It's possible to integrate the attitude control with the trajectory controller once it's designed and optimized. The control architecture to be implemented on the quadcopter system is shown in Figure 3.

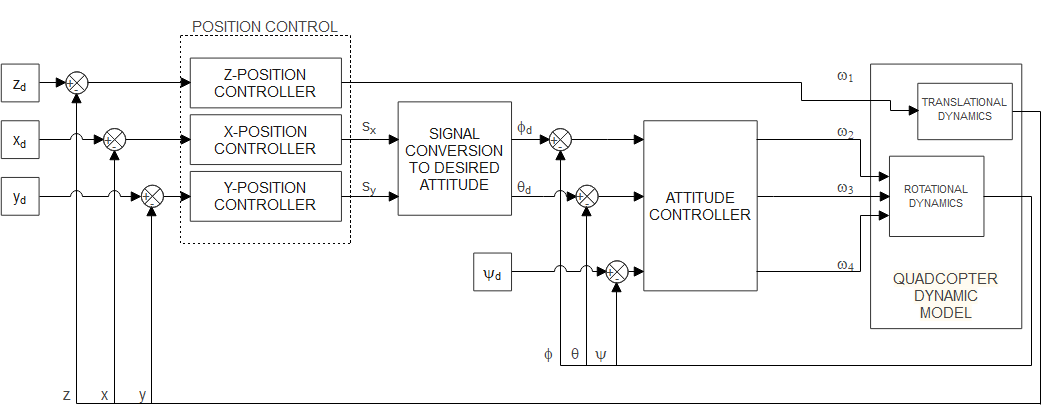


Figure 3: Control Architecture

**PID Control**

The general form of a PID controller is given as:

where u(t) represents the control input, r(t) represents the desired state and y(t) is the current or actual state. KP, KI and KD are the gain parameters for the proportional, integral and derivative terms of the PID controller.

The proportional and derivative terms would be used for the quadcopter control. The torque generated is proportional to the angular velocities therefore we set the torques to be proportional to the controller output such that:

The input to the quadcopter system is the angular velocity of the rotors. Recall equation 20 which relates the torque to the square of the angular velocity of the rotors, there are three equations but four unknowns. To allow simplification, the total thrust which affects the acceleration in the z-direction is set to be equal to . This constraint is enforced to keep the quadcopter flying. Converting this thrust equation to the appropriate reference frame and utilizing a PD to minimize the error in the z-axis:

Solving for the angular velocities of the rotor by computing equation 32 and equating it to equation 20:

Simplifying equation 34 would give the equations:

**Feedback Linearization**

Feedback linearization is a nonlinear control technique that has sparked a lot of interest in recently. The main concept is to mathematically convert nonlinear system dynamics into (fully or partially) linear ones, allowing linear control methods to be used. The goal is to design a controller such that it exactly cancels out the system dynamics.

From equation 28, we can replace , , where , , and are gain values and replace accordingly.

Expanding equation 39 above would give:

Taking the square of both sides and adding equation 40, 41 and 42 together to get:

To obtain , from the x and y axes portion of equation 39, let:

From equation 45:

Substitute equation 46 into equation 44:

To obtain , from the x and z axes portion of equation 3.39, let:

Substituting equation 49 and 52 into equation 44:

Recall:

For equation 3.50 and equation 3.60, and are given as:

From equation 8 and 32 in terms of desired angles, we can replace , where is the gain values and replace and accordingly:

The angular velocities and can be obtained using equation 34.

**C Trajectory Tracking**

The goal of trajectory control is to take the quadcopter system from its current position to the desired position by regulating the quadcopter's rotor angular velocities. Due to its complex dynamics, finding the best quadcopter trajectory is a huge challenge.

*Trajectory Tracking with PD Control*

To relate the desired position x and y which are not controllable to the desired angles of and that are controllable, we make two assumptions:

1. Small angle approximation such that and .
2. The desired angle is zero.

From the above assumptions, equation 3.28 can be simplified as:

Equations 3.39, 3.40 and 3.41 can be rewritten as:

, and represent the desired values of pitch angle, roll angle and thrust.

*Trajectory Tracking with FBL Control*

To relate the desired position x and y which are not controllable to the desired angles of and that are controllable using FBL control techniques, we make two assumptions:

1. No small angle approximation
2. The desired angle is zero.

**IV Simulation and Results**

**A Simulation**

The dynamic model and controllers are implemented in MATLAB 2021 for simulation with MATLAB programming language.

Table 1: Quadcopter parameter values for simulation

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Unit** |
| g | 9.81 | m/s2 |
| m | 0.468 | kg |
| l | 0.225 | m |
| Jr |  | kg m2 |
| kt |  |  |
| kb |  |  |
| Ixx |  | kg m2 |
| Iyy |  | kg m2 |
| Izz |  | kg m2 |
| kdx | 0.25 |  |
| kdy | 0.25 |  |
| kdz | 0.25 |  |

The quadcopter model is simulated with parameter values as shown in Table 1 [4]. The initial conditions assigned to the system for simulation is given in Table 2.

Table 2: Initial conditions for simulation

|  |  |  |  |
| --- | --- | --- | --- |
| **State** | **Value** | **State** | **Value** |
| x | -1 | ϕ | 10 |
| y | 2 | θ | -10 |
| z | 1 | ψ | 5 |

The simulation progresses at 0.001 second intervals to a total time of 15 seconds. The control inputs (angular velocities of the rotors) are shown in Figure 4, the positions and the angles in Figure 5.

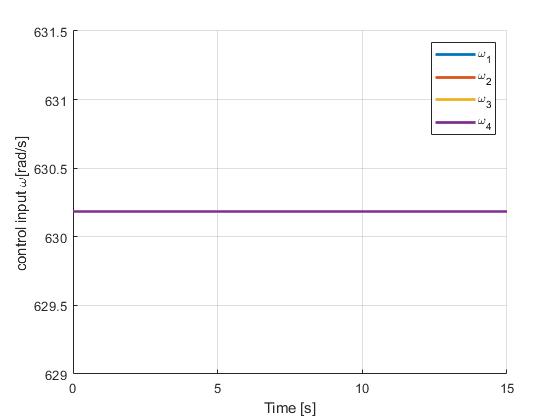


Figure 4: Control input

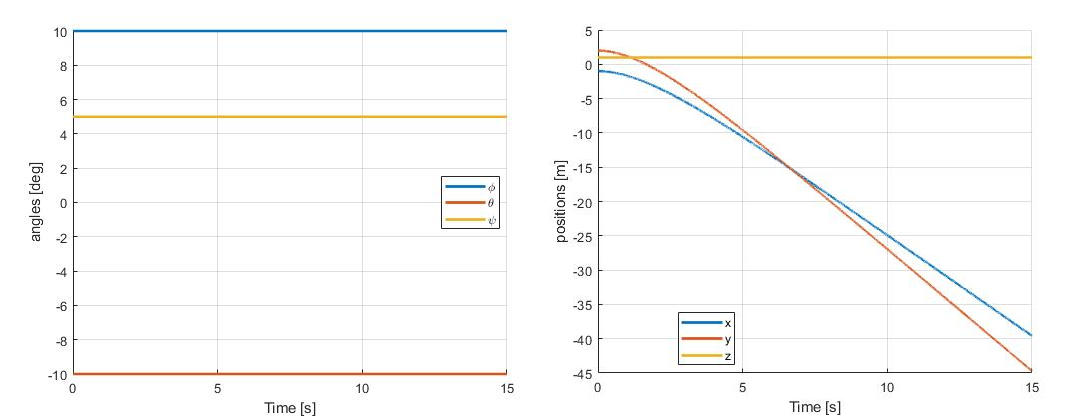


Figure 5: Position and angle variables against time

The control input remains constant and hence the angles remain the same but the positions become unstable.

**Attitude Stabilization**

A PD controller is implemented on the nonlinear system to stabilize and drive the system states to zero. The initial conditions assigned for the system remain the same as shown in Table 2. The control gain values for the PD controller are determined by manual tuning and are shown in Table 3.

Table 3: Gain values for PD controller

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Value** | **Parameter** | **Value** |
|  | 1.5 |  | 2.6 |
|  | 1.5 |  | 2.6 |
|  | 1.5 |  | 2.6 |
|  | 6 |  | 1.5 |

The simulation progresses at 0.001 second intervals to a total time of 15 seconds. The angles are shown in Figure 6, the angular velocities of the rotors are shown in Figure 7, and the positions in Figure 8. The angles are stabilized to zero after 10 seconds and the position z is stabilized to zero after 5 seconds. The and positions do not stabilize to zero because the states are not observable.

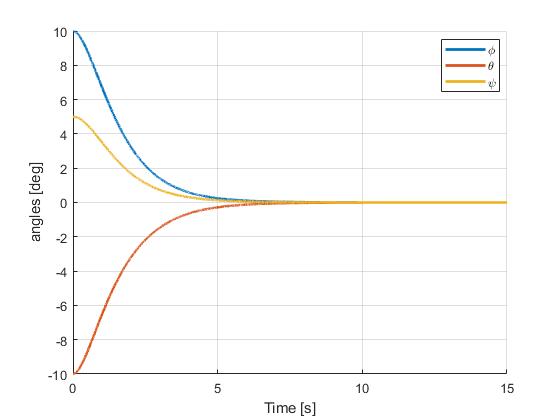


Figure 6: Angles

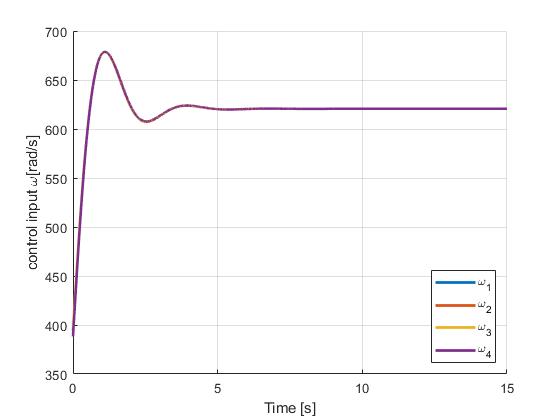


Figure 7: Control input

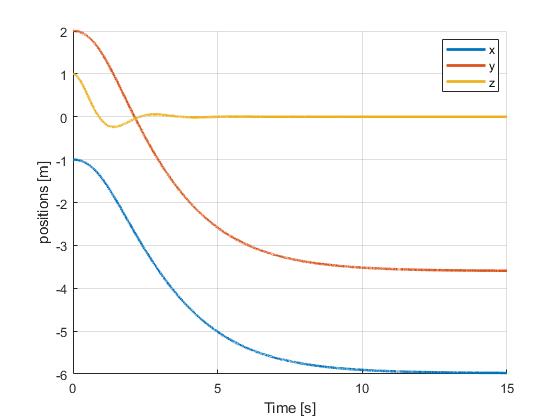


Figure 8: Positions x, y and z

Using the concept of trajectory tracking, an attempt can be made to drive the and positions to zero. This can be done by mapping the zero desired states of and to the desired and states which are controllable as shown in equation 3.42, 3.43 and 3.44. An additional proportional controller , with a control gain of 0.04 is implemented to drive the system states to zero and the simulation advances at 0.001 second intervals to a total time of 45 seconds. The angular velocities of the four rotors and the angles are shown in Figure 9, the modified positions and are shown in Figure 10.

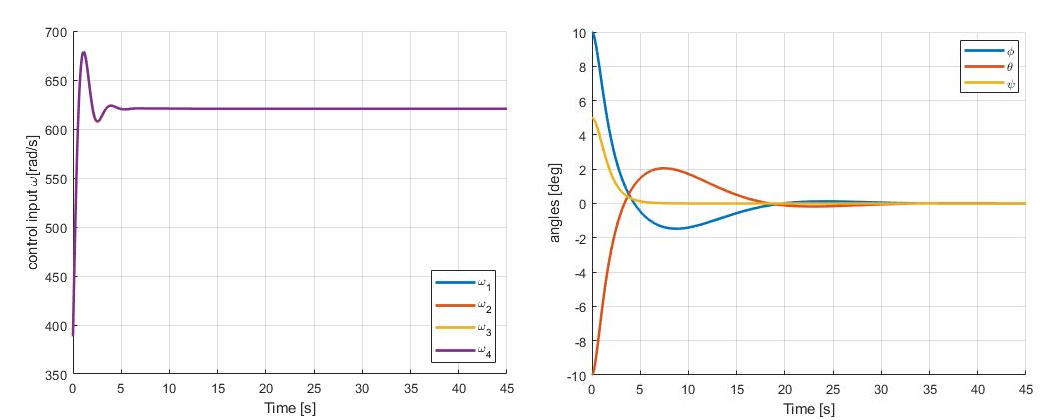


Figure 9: Control input and angles

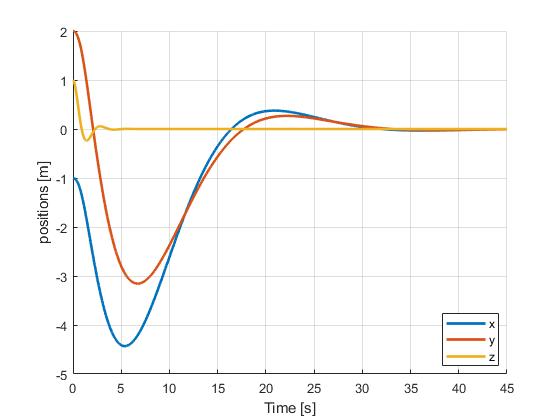


Figure 10: Positions x, y and z

The angles are stabilized to zero after 35 seconds and the positions and are stabilized to zero after 45 seconds.

**Trajectory Tracking**

Two controls for trajectory tracking are considered and simulated for two different trajectories. The first control strategy is implemented with a PD controller and the second strategy is a PD controller integrated with the feedback linearized system. The total time assigned for the simulation in both cases is 45 seconds.

*Trajectory Tracking with PD Controller*

The comparison between the desired and actual trajectory for the first path is shown in Figure 11. Figure 12 shows the graph of the position variables against time.

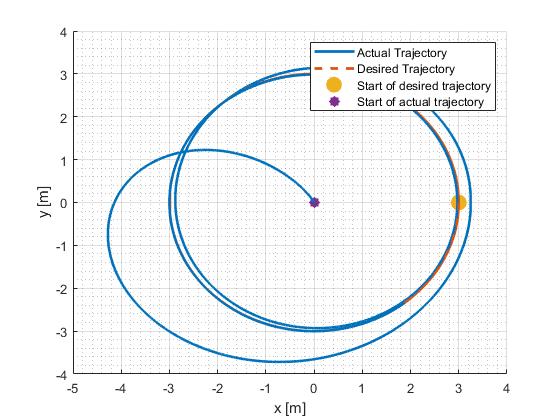


Figure 11: Desired and actual trajectory

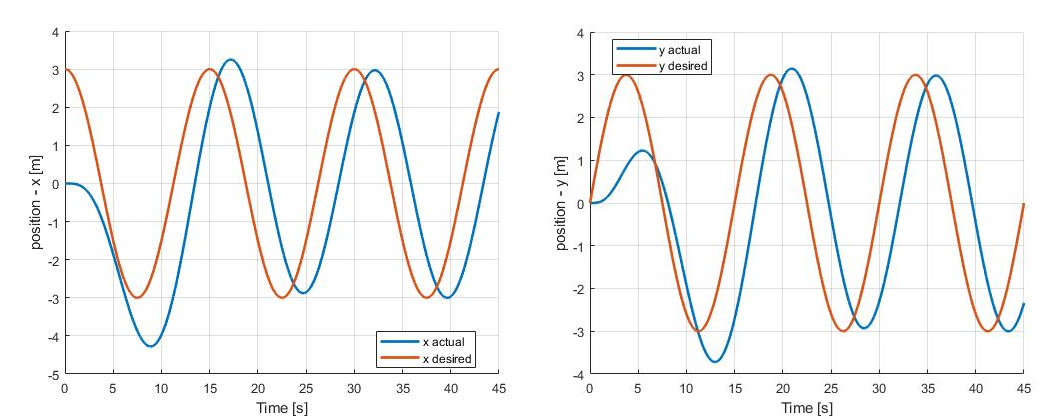


Figure 12: Coordinate wise comparison between desired and actual trajectory

Another trajectory path is simulated and the result is shown in Figure 13. The graph of the position variables against time for this path is shown in Figure 14.

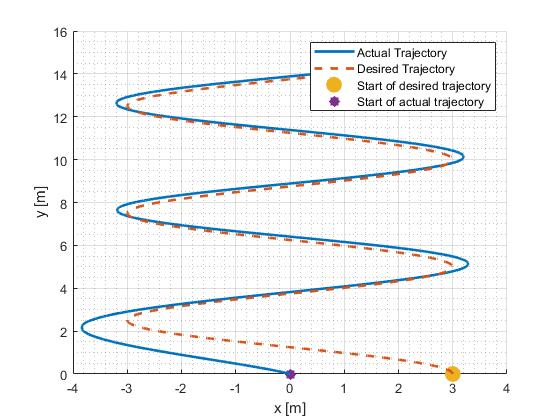


Figure 13: Desired and actual trajectory

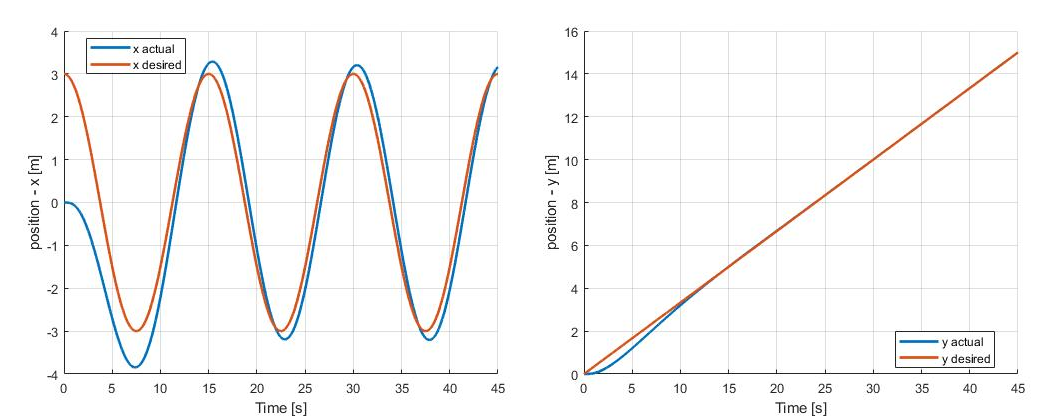


Figure 14: Coordinate wise comparison between desired and actual trajectory

A similar result as seen in Figure 12 is also obtained; the quadcopter is unable to follow the desired trajectory with precision.

*Trajectory Tracking with Feedback Linearization and PD Controller*

Feedback linearization technique is implemented on the quadcopter model and simulated with a PD controller. The desired trajectory and the actual trajectory for the first path is shown in Figure 15. Figure 16 shows the graph of the position variables against time.

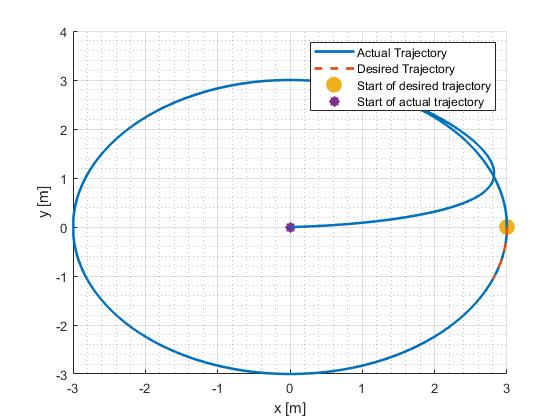


Figure 15: Desired and actual trajectory

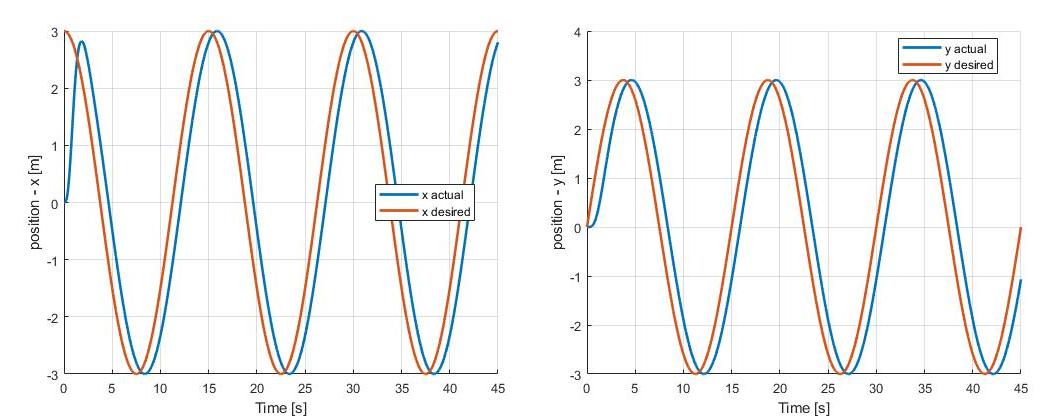


Figure 16: Coordinate wise comparison between desired and actual trajectory

Another trajectory path is simulated and the result is shown in Figure 17. A similar result as seen in Figure 15 is also obtained; the quadcopter is able to lock on to the desired trajectory with more accuracy with this control.

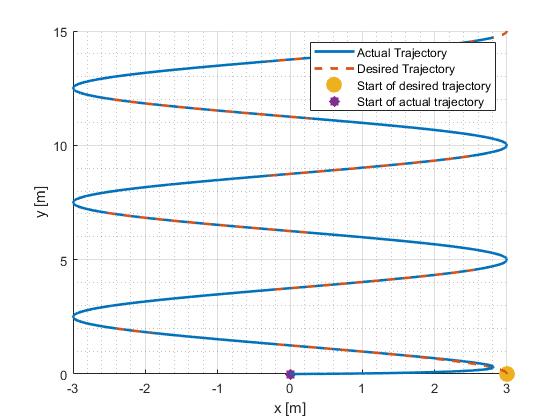


Figure 17: Desired and actual trajectory

**B Result Analysis**

A detailed analysis on the performance of the PD controller for stabilization and its integration with the FBL system is required to determine the most suitable control strategy.

**Attitude Stabilization**

As discussed earlier, the PD controller with control gains shown in Table 3 was able to stabilize the angles to the desired zero state after 10 seconds and also the drive the z position variable to zero as well in 5 seconds. The x and y positions are eventually stabilized but they do not stabilize to the desired zero state. By mapping the desired states of and to the desired and states which are controllable, an additional proportional controller is used to drive all the system states to zero. However, this approach is not efficient because it takes too long for the angles and position to stabilize. The angles and stabilize to zero after 35 seconds and the positions and stabilize to zero after 45 seconds. The and states experience overshoots before stabilizing to zero. A side-by-side comparison of the angle and position variables of the two approaches are shown in Figure 18 and 19.

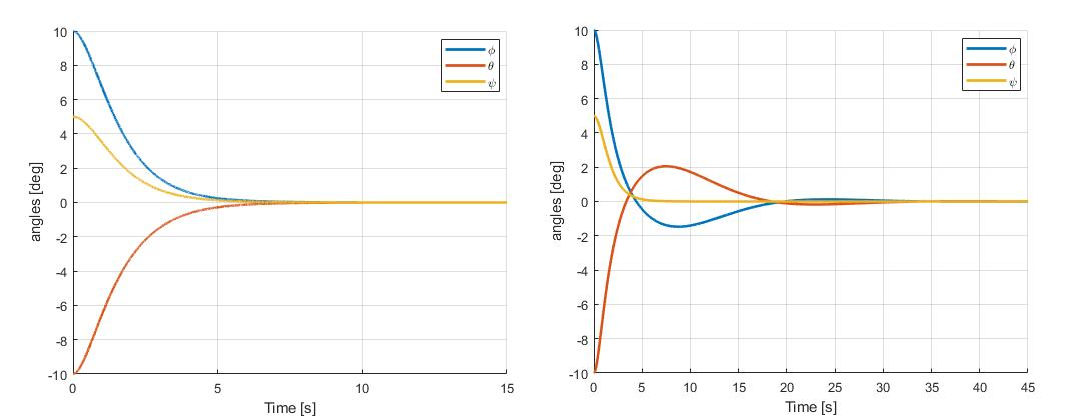


Figure 18: Angles

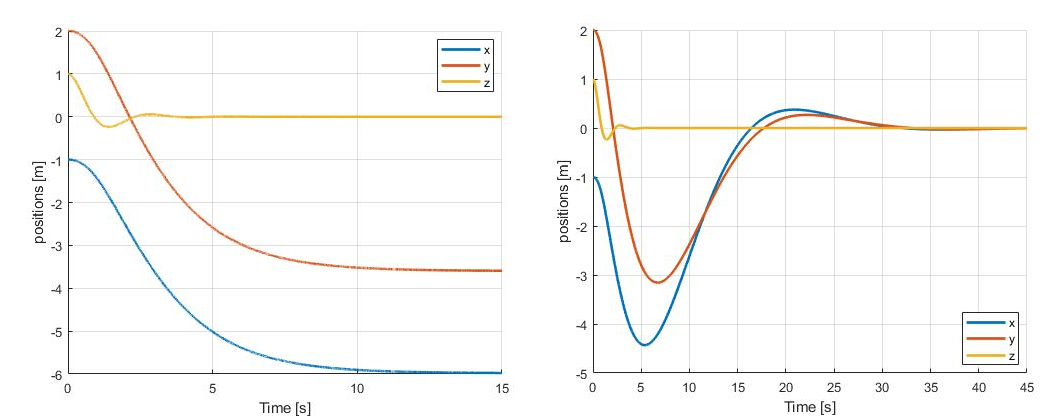


Figure 19: Positions x, y and z

**Trajectory Tracking**

The results of the two control strategies implemented for trajectory tracking were previously discussed. A better performance is observed in the feedback linearized system with PD control. The quadcopter is able to lock on to the desired trajectory and follow the path with accuracy in a lesser time period. Figure 20, 21 and 22 show the plot of the desired and actual trajectory after 15, 30 and 45 seconds respectively for both the PD control and the FBL with PD control.

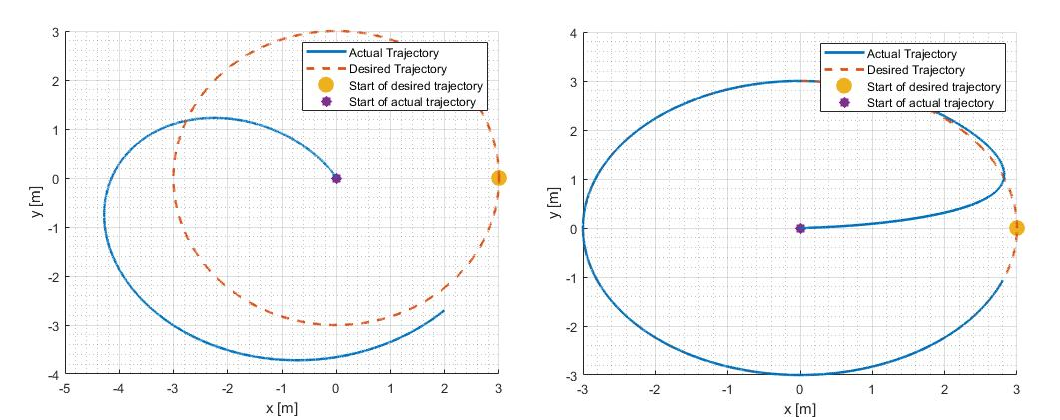


Figure 20: Trajectory for PD (left) and FBL with PD control (right) after 15 seconds

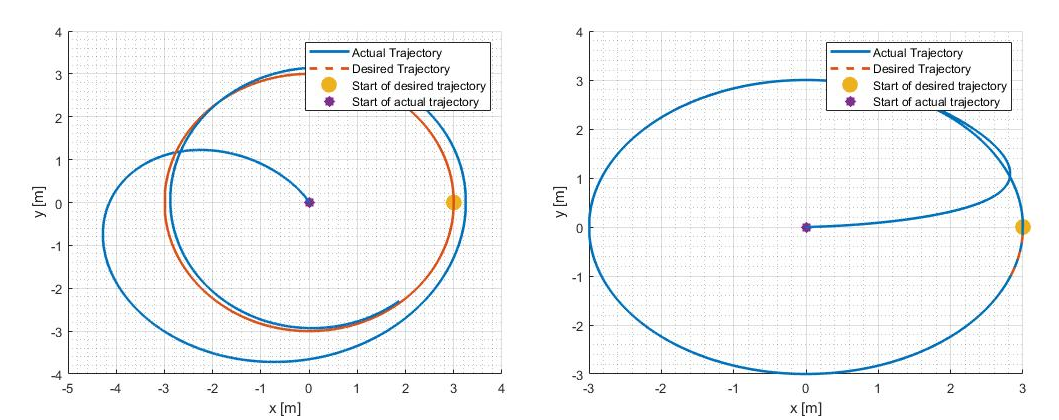


Figure 21: Trajectory for PD (left) and FBL with PD control (right) after 30 seconds

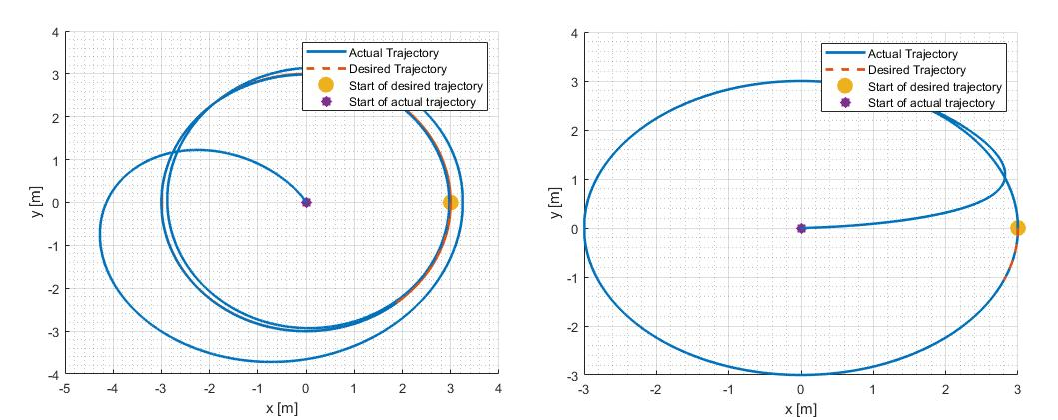


Figure 22: Trajectory for PD (left) and FBL with PD control (right) after 45 seconds

The quadcopter system simulated with the FBL and PD control is able to lock on to the desired trajectory after the first 15 seconds while the system with only the PD control is unable to do that properly even after 30 seconds. This shows that the feedback linearization technique with the PD control is more effective in trajectory tracking than the regular PD control.

**V Conclusion**

The mathematical modelling of the quadcopter system dynamics has been developed based on a few standard assumptions. This was achieved with proper understanding of the different coordinate systems involved, Euler angles and the Newtonian equation to express the various extraneous forces acting on the quadcopter body. The state space model of the system dynamics was obtained using Newton’s and Euler’s laws. The model was tested by simulating the quadcopter flight on MATLAB with already established parameters.

Attitude stabilization and control of the quadcopter system has been achieved by using a PD controller. The simulation showed that the PD control algorithm was able to stabilize the system states to the desired states. However, the PD controller could not stabilize the and positions to the desired state because the state variables are not observable. Another mapping approach was considered which was able to drive the x and y positions to the desired zero state but this was achieved after a longer period of time.

Two nonlinear control strategies were implemented for trajectory tracking of the quadcopter system. The simulation showed that the PD control was inferior to the FBL with PD control as the latter approach presented accurate tracking of the desired trajectory in a lesser period of time.

The NDI technique used in feedback linearization of the system is implemented based on the assumption that all the states are available and measurable. Since this does not always seem to be the case, a further step in this control strategy can be the design of a suitable estimator or observer which can generate the possible outcomes of a full state from partial state variables. Another step forward in this research is the design of a controller to address the failure of one or more rotors. The quadcopter must be configured to apply the contingency algorithm once failure is detected based on inputs of the controller. Also, the proposed model and control techniques were evaluated using simulations only. Actual prototypes of a quadcopter should be built to obtain more practical and accurate results.

**References**

[1] M. Etemadi, “Mathematical dynamics, kinematics modeling and PID equation controller of quadcopter,” *Int. J. Appl. Oper. Res. Open Access J.*, vol. 7, no. 1, pp. 77–85, 2017.

[2] S. Bouabdallah, “Design and control of quadrotors with application to autonomous flying,” Epfl, 2007.

[3] F. Sabatino, “Quadrotor control: modeling, nonlinearcontrol design, and simulation.” 2015.

[4] T. Luukkonen, “Modelling and control of quadcopter,” *Indep. Res. Proj. Appl. Math. Espoo*, vol. 22, p. 22, 2011.

[5] A. Zulu and S. John, “A review of control algorithms for autonomous quadrotors,” *arXiv Prepr. arXiv1602.02622*, 2016.

[6] E. Gopalakrishnan, “Quadcopter flight mechanics model and control algorithms,” *Czech Tech. Univ.*, vol. 69, 2017.